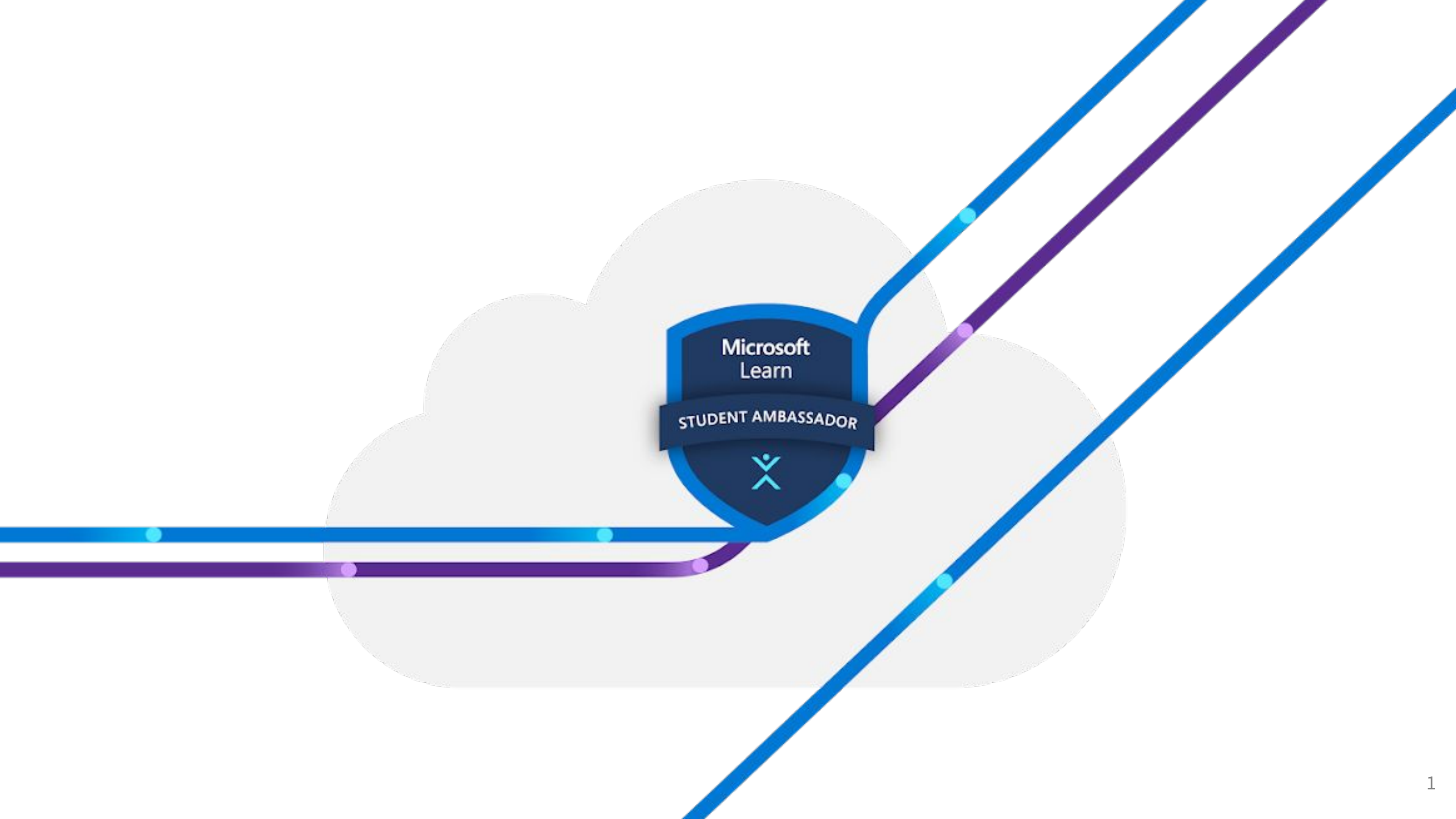

Microsoft
Learn
STUDENT AMBASSADOR

Microsoft Learn
Student Ambassadors

Git & Github For Beginers

GOURAV KUSHWAHA

Microsoft Learn
Student Ambassadors

Basics to Git, Github, Versioning Systems

GOURAV KUSHWAHA
BETA MLSA

Introduction to Git & GitHub & Version Control

Presented by: GOURAV KUSHWAHA

Agenda

- What is Version Control?
- It’s time to get Git.
- Working with repositories locally.
- Working with remotes.
- Working on another developer’s repository.

What is Version Control?

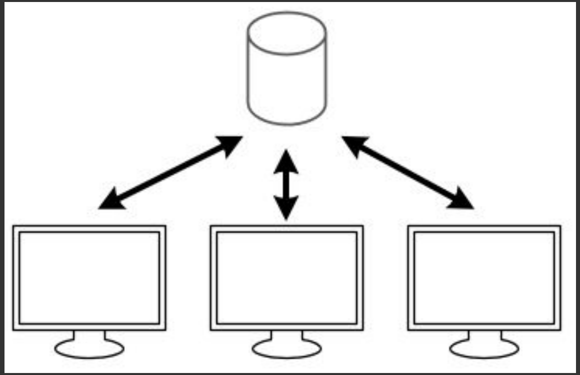
- Time
- Who took it
- State and Location of pieces

- Time
- Who took it
- State and Location of pieces

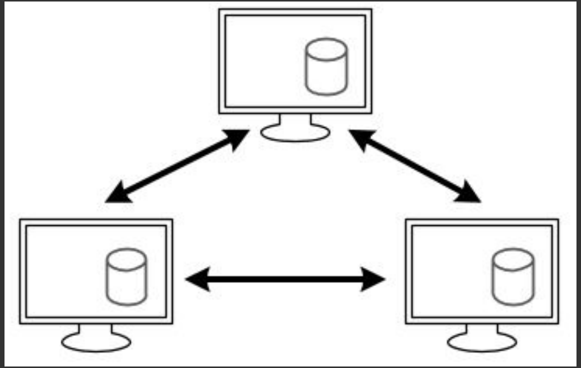
This Idea of a safe point is exactly What Version Control is

Version Control Systems (VCS)

Centralized
(CVCS)



Distributed
(DVCS)



Most Popular Version Control Systems

- Git
- Subversion
- Mercurial

git

Distributed Version Control System.

git

Version Control Tool

GitHub

Service hosts Git Projects

Terminology

Branching

Merging

Version Control System

SHA

Staging Index

Merge Conflicts

Source Code Manager

Index

Remotes

Staging area

Repository

Commit

Working Directory

Repo

Checkout

Terminology

git-cheat-sheet-education (github.com)

Exercise –
Install and Configure Git

In this exercise, we will:

- Download and install Git.
- Perform first time configuration for Git.
- Configure Git with out Code Editor.

How to get Git?

Installing Git

MacOS / Windows

1. go to git-scm.com/downloads
2. download the software for Mac/Windows
3. install Git choosing all of the default options

Linux

```
$ sudo apt-get install git
```

First Time Git Configuration

```
# sets up Git with your name
git config --global user.name "<Your-Full-Name>"

# sets up Git with your email
git config --global user.email "<your-email-address>"

# makes sure that Git output is colored
git config --global color.ui auto
```

Git & code Editor

Atom Editor Setup

```
git config --global core.editor "atom --wait"
```

VSCode Setup

```
git config --global core.editor "code --wait"
```

Review Git Configuration

```
# lists all the configuration properties
git config --list
```

Exercise –
Working with a Local Repository

In this exercise, we will:

- Create a Git Local Repository.
- Make Changes, add, and commit them.
- Review the repository’s history.

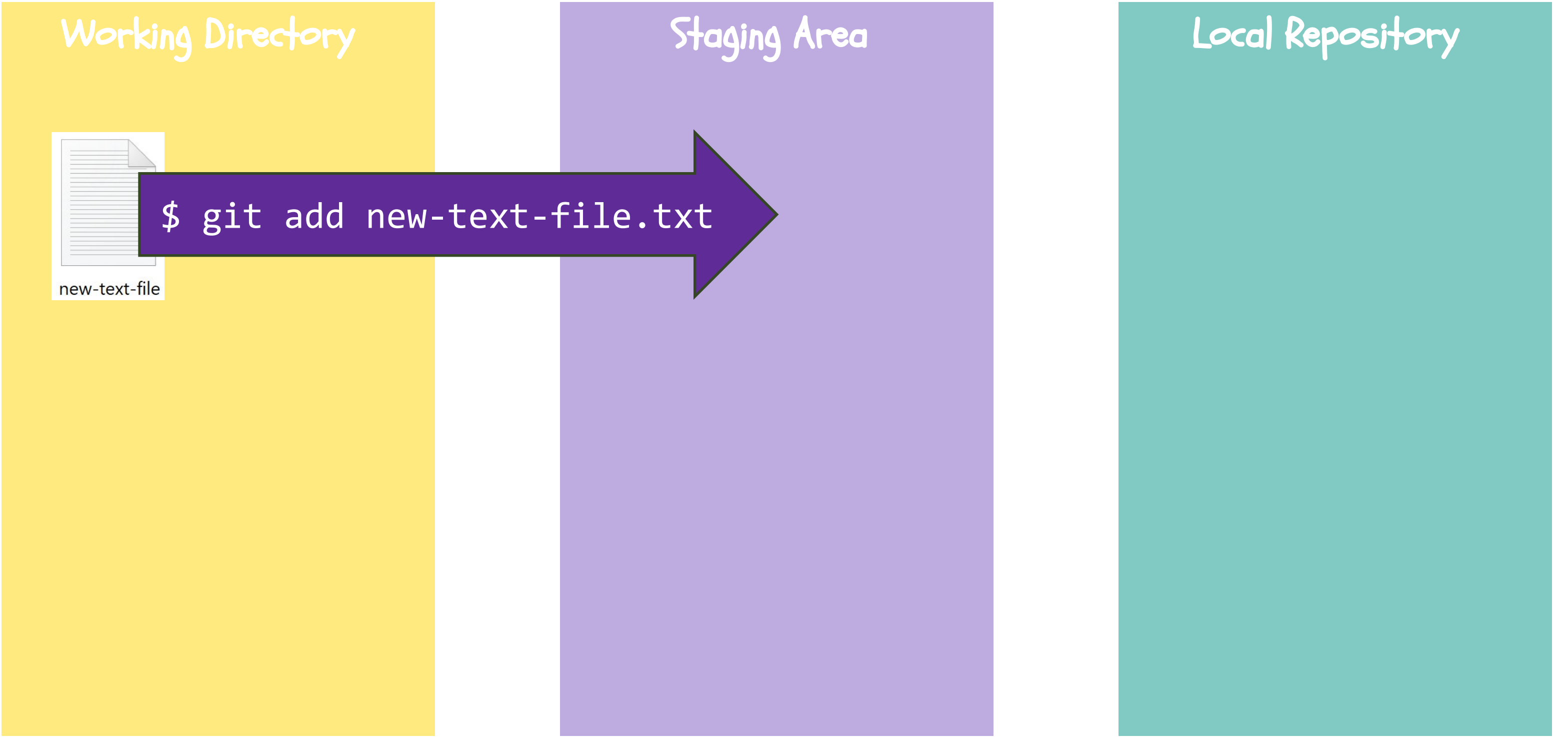
Initializing a Local Git Repository

```
$git init
```

Common terminal commands

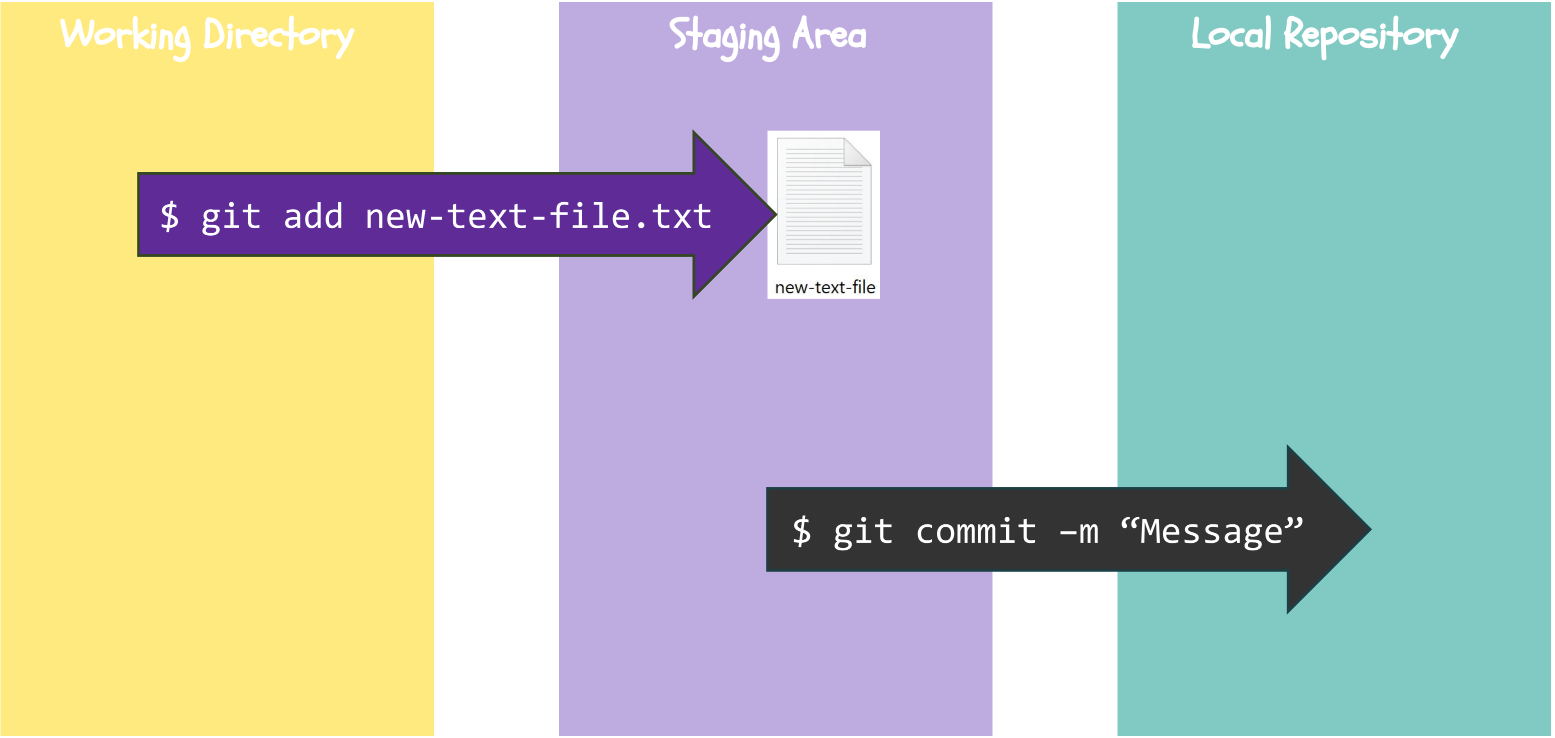
```
$ls    - used to list files and directories
$mkdir - used to create a new directory
$cd    - used to change directories
$rm    - used to remove files and directories
$pwd   - used to print working directory
$touch - used to create and modify files
$start - used to open files or directories
```

How Git works?



```
$ git add new-text-file.txt
```

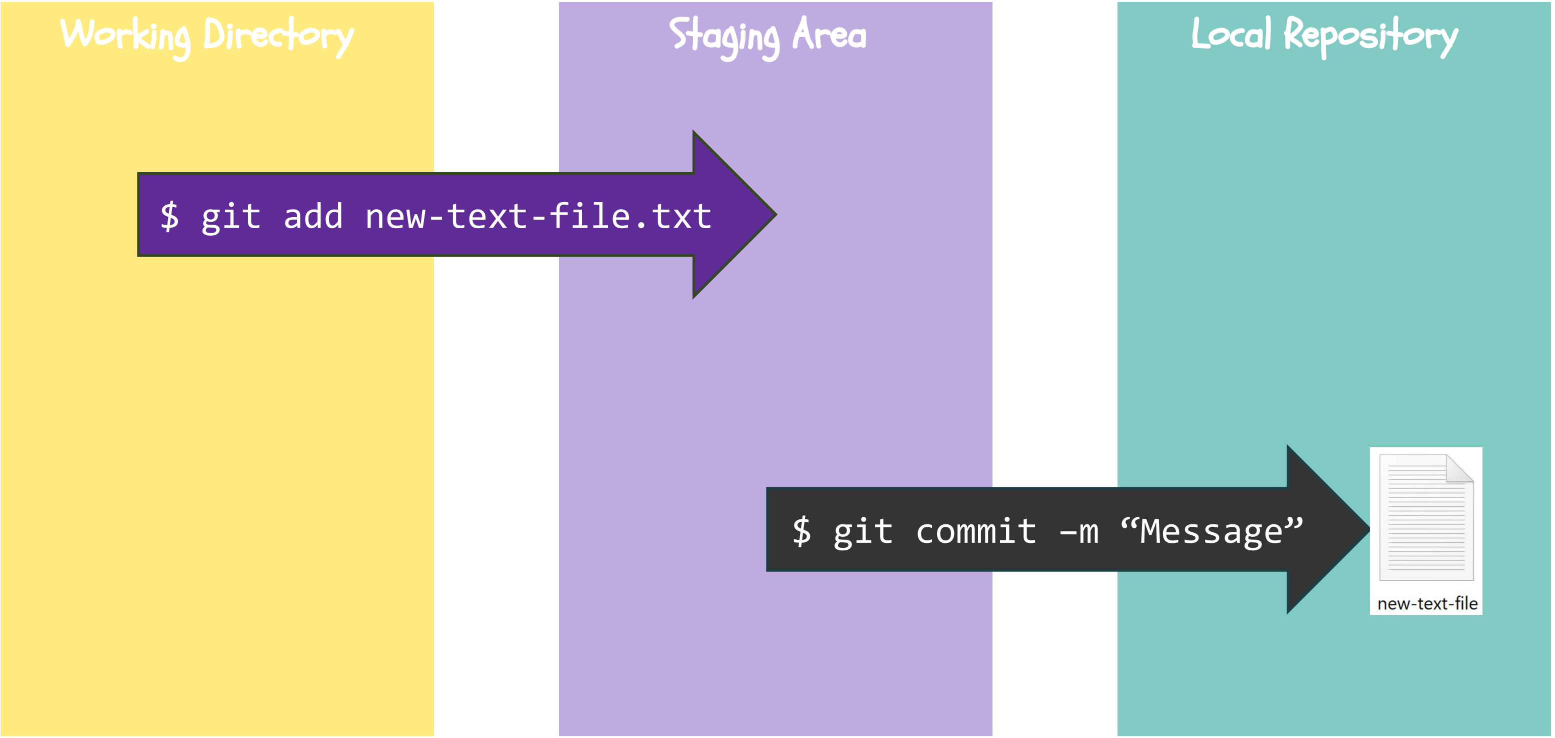
How Git works?



```
$ git add new-text-file.txt
```

```
$ git commit –m “Message”
```

How Git works?



```
$ git add new-text-file.txt
```

```
$ git commit –m “Message”
```

Git Add

```
$ git add <file1> <file2> … <fileN>

$ git add .
```

This command moves your changes to the staging area.

Git Commit

```
$ git commit -m “Initial commit”
```

This command moves your changes to the local repository.

Try to always write your commit message in an imperative way.

Git Commit Message

Do’s

- Keep the message short (less than 60-ish characters)
- Explain what the commit does (not how or why!)

Don’ts

- Explain why the changes are made
- Explain how the changes are made
- Use the word “and”

Reviewing the Repository’s History

```
$ git status
```

```
On branch master
Your branch is up-to-date with 'origin/master'.
Nothing to commit, working directory clean
```

```
$ git log
```

Reviewing the Repository’s History

```
$ git status
```

This command display’s the status of the working directory and the staging area.

It doesn’t show us history!

Reviewing the Repository’s History

```
$ git log
```

This command will show you the history of changes in the repository including:

- the SHA
- the author
- the date
- the commit message

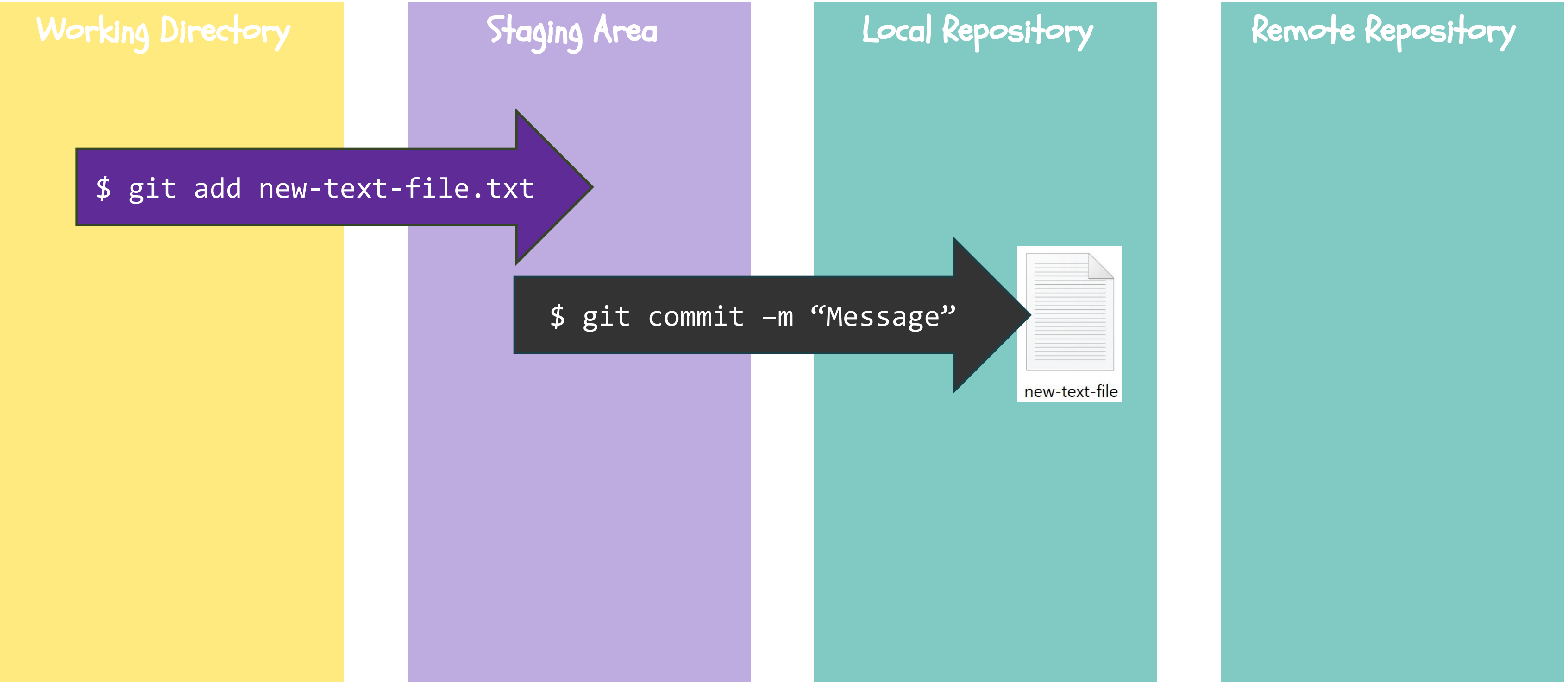
```
$ git log --oneline
```

Exercise –
Working with Remotes

In this exercise, we will:

- Create a GitHub repository and clone it.
- Make changes and push to GitHub.
- Make changes and pull from GitHub.
- Fork a repository and make a pull request.

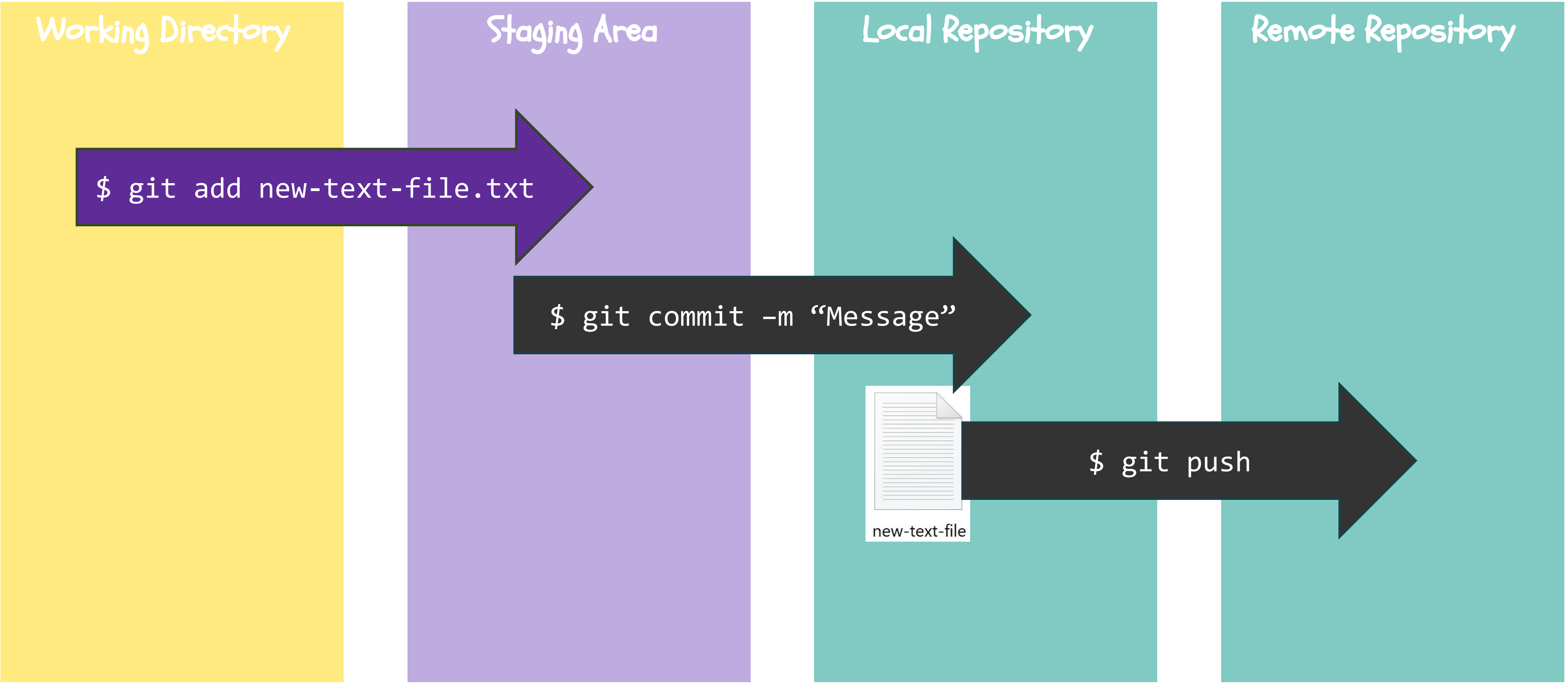
How Git works with GitHub?



```
$ git add new-text-file.txt
```

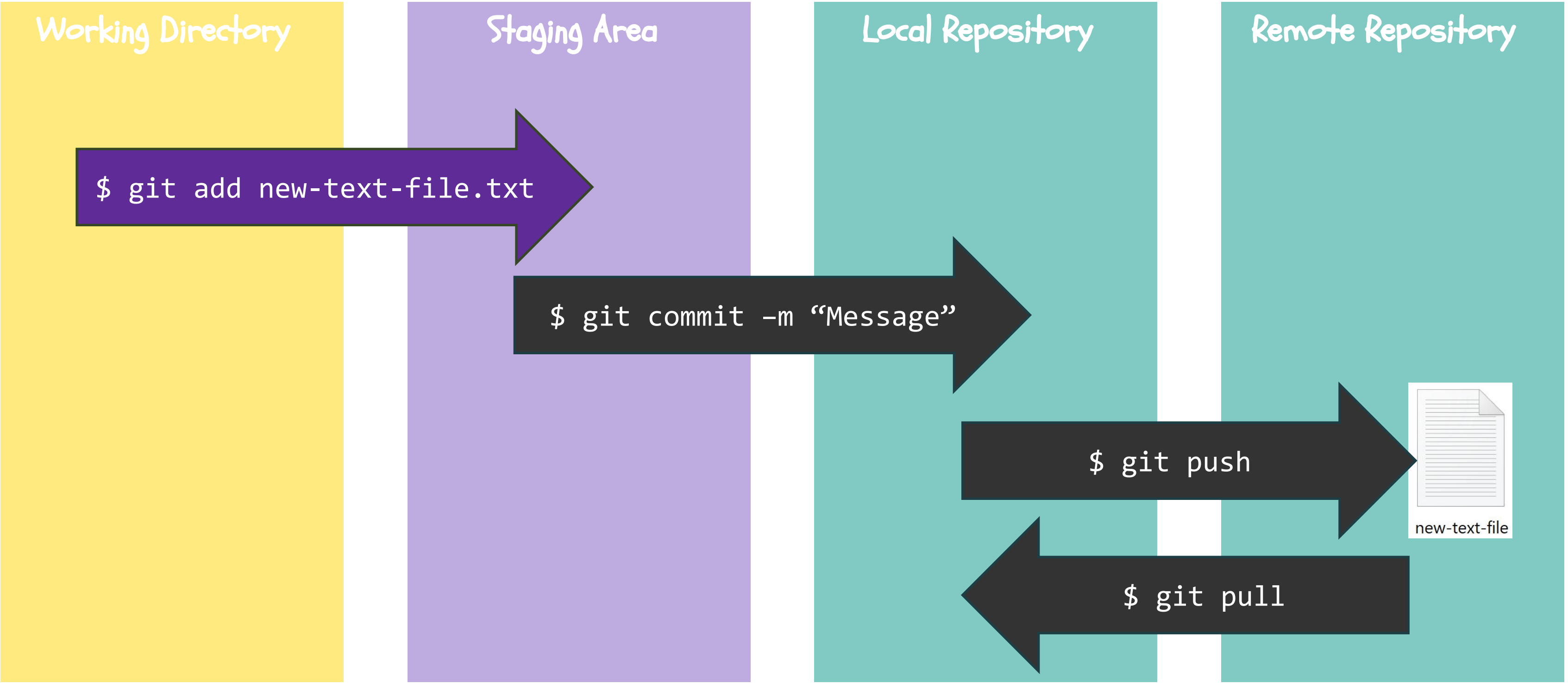
```
$ git commit -m “Message”
```

How Git works with GitHub?



```
$ git add new-text-file.txt
$ git commit -m “Message”
$ git push
```

How Git works with GitHub?



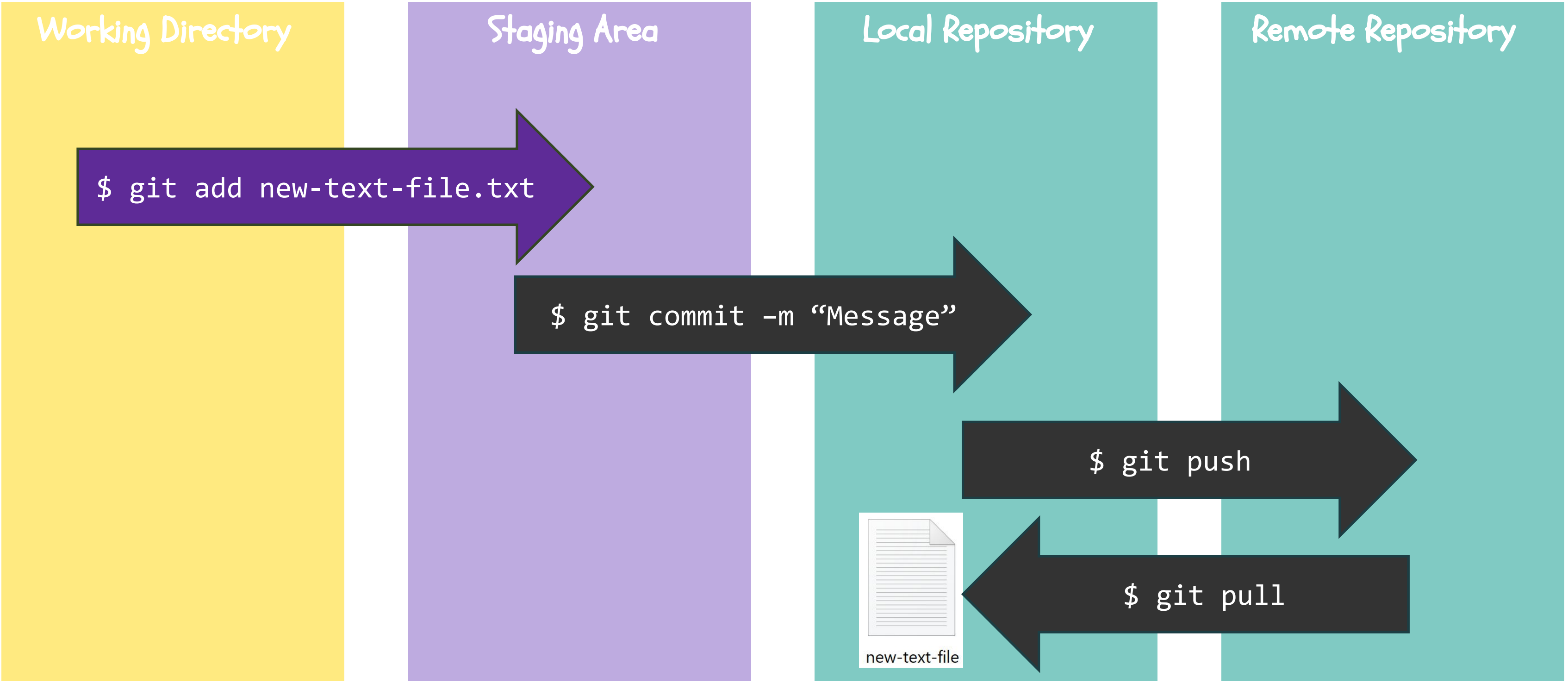
```
$ git add new-text-file.txt
```

```
$ git commit -m “Message”
```

```
$ git push
```

```
$ git pull
```

How Git works with GitHub?



```
$ git add new-text-file.txt
$ git commit -m “Message”
$ git push
$ git pull
```

Next Steps

- Learn more about GitHub on Microsoft Learn.
- Finish Introduction to version control with Git learning path.

Please tell us how you liked this workshop by filling out this survey:

https://aka.ms/workshopomatic-feedback